

Finite Difference Schemes and the Schrodinger Equation

Jonathan King, Pawan Dhakal

June 2, 2014

1 Introduction

In this paper, we primarily explore numerical solutions to the Quantum 1D Infinite Square Well problem, and the 1D Quantum Scattering problem. We use different finite difference schemes to approximate the second derivative in the 1D Schrodinger's Equation and linearize the problem. By doing so, we convert the Infinite Well problem to a simple Eigenvalue problem and the Scattering problem to a solution of a system of linear equations. We examine the convergence of the solution to the infinite square well problem for high order stencils, and compare the computed results to an analytic solution. For the scattering problem, we test both first and second order finite difference schemes for boundary conditions, and compare the convergence of these schemes.

2 Finite Difference Schemes

Finite Difference Methods are used to approximate derivatives to solve differential equations numerically. The goal is to discretize the domain of the given problem, for example the x grid for a function $f(x)$, and use the value of the function evaluated at a point and neighbouring point(s) to approximate the derivative of the function at the point. The assumption made is that the functions to be differentiated are well behaved. Then using truncated Taylor's sum, one can derive approximations for derivatives. In our project, we needed approximations for the first derivative and the second derivative.

2.1 Stencils and Differences

Depending on the number of neighbouring points used in approximate derivatives, we can define our approximation as being the 'three point stencil' if we are using two neighbouring points along with the evaluation point, or as the 'five point stencil' if we are using four neighbouring points along with the evaluation point and so on.

Similarly, we can define our approximations as being centered difference, backward difference, or forward difference depending upon the symmetry of the neighbouring points. For example: $u' \approx \frac{u_{i+1} - u_i}{\Delta x}$ is the forward (two point stencil) finite difference method for approximating first derivative of u . Similarly, $u' \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}$ is the centered finite difference method, and $u' \approx \frac{u_i - u_{i-1}}{\Delta x}$ is the backward finite difference method.

2.2 Deriving Coefficients and Accuracy

To derive the coefficients for finite difference, we use truncated Taylor series to approximate the value of the function at the point of evaluation of the derivative and at the neighbouring points. We can use the leading term in truncated error (i.e. real sum - truncated sum) to approximate the error in the truncated Taylor series and predict the accuracy of the approximate derivative. Using a general equivalence relation $f^n(x) \approx \dots + af(x-h) + bf(x) + cf(x+h) + \dots$ depending upon the no of stencils and the type of difference (backward, forward, or centered) we want, we can find the coefficients a, b, c, \dots to approximate the n th derivative of f .

For the purposes of this project, we need the following approximations:

1. Centered Five Point Stencil for Second Derivative

Use $f''(x) = af(x - 2h) + bf(x - h) + cf(x) + df(x + h) + ef(x + 2h)$.

$$\begin{aligned} f(x - h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) + O(h^5) \\ f(x + h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) + O(h^5) \\ f(x - 2h) &= f(x) - 2hf'(x) + \frac{4h^2}{2}f''(x) - \frac{8h^3}{6}f'''(x) + \frac{16h^4}{24}f^{(4)}(x) + O(h^5) \\ f(x + 2h) &= f(x) + 2hf'(x) + \frac{4h^2}{2}f''(x) + \frac{8h^3}{6}f'''(x) + \frac{16h^4}{24}f^{(4)}(x) + O(h^5) \end{aligned}$$

Equating Coefficients:

$$\begin{aligned} f''(x) &= af(x - 2h) + bf(x - h) + cf(x) + df(x + h) + ef(x + 2h) \\ a + b + c + d + e &= 0 \\ -2a - b + d + e &= 0 \\ 2h^2a + h^2/2b + h^2/2d + 2h^2e &= 1 \\ -4/3a - 1/6b + 1/6d + 4/3e &= 0 \\ 2/3a + 1/24b + 1/24d + 2/3e &= 0 \\ \implies a = -\frac{1}{12h^2}; b = \frac{4}{3h^2}; c = -\frac{5}{2h^2}; d = \frac{4}{3h^2}; e = -\frac{1}{12h^2} \end{aligned}$$

To find error:

$$f''(x) = -\frac{1}{12h^2}f(x - 2h) + \frac{4}{3h^2}f(x - h) + -\frac{5}{2h^2}f(x) + \frac{4}{3h^2}f(x + h) + -\frac{1}{12h^2}f(x + 2h)$$

Taylor's theorem:

$$\begin{aligned} af(x - 2h) &= af(x) - 2ahf'(x) + \frac{4ah^2}{2}f''(x) - \frac{8ah^3}{6}f'''(x) + \frac{16ah^4}{24}f^{(4)}(x) - \frac{32ah^4}{120}f^{(5)}(x) + \frac{64ah^6}{6!}f^{(6)}(a_1) \\ ef(x + 2h) &= ef(x) + 2ehf'(x) + \frac{4eh^2}{2}f''(x) + \frac{8eh^3}{6}f'''(x) + \frac{16eh^4}{24}f^{(4)}(x) + \frac{32eh^4}{120}f^{(5)}(x) + \frac{64eh^6}{6!}f^{(6)}(a_2) \end{aligned}$$

Adding:

We have: $a = e$

$$af(x - 2h) + af(x + 2h) = 2af(x) + 4ah^2f''(x) + \frac{4ah^4}{3}f^{(4)}(x) + \frac{64ah^6}{6!}(f^{(6)}(a_1) + f^{(6)}(a_2))$$

Similarly:

We have: $b = d$

$$\begin{aligned} bf(x - h) + bf(x + h) &= 2bf(x) + bh^2f''(x) + \frac{bh^4}{12}f^{(4)}(x) + \frac{bh^6}{6!}(f^{(6)}(a_3) + f^{(6)}(a_4)) \\ \therefore f''(x) &= \hat{f}''(x) + \left(-\frac{1}{135}(f^{(6)}(a_1) + f^{(6)}(a_2)) + \frac{1}{540}(f^{(6)}(a_3) + f^{(6)}(a_4)) \right) h^4 \end{aligned}$$

where, $a_1, a_2, a_3, a_4 \in (x, x + h)$.

As $h \rightarrow 0$, the term in the bracket \rightarrow a constant

$$\therefore f''(x) = \hat{f}''(x) + O(h^4)$$

2. **Assymmetric Forward Five Point Stencil for Second Derivative:** We wish to find an approximation for $f''(x)$ as the following non-symmetric five point stencil:

$f''(x) = af(x-h) + bf(x) + cf(x+h) + df(x+2h) + ef(x+3h)$ Carrying out the process outline above for the centered case, we derived the following coefficients:

$$a = -\frac{11}{12h^2}; b = \frac{-5}{3h^2}; c = \frac{1}{2h^2}; d = \frac{1}{3h^2}; e = -\frac{-1}{12h^2}$$

3. **Assymmetric Backward Five Point Stencil for Second Derivative:** We wish to find an approximation for $f''(x)$ as the following non-symmetric five point stencil:

$f''(x) = af(x+h) + bf(x) + cf(x-h) + df(x-2h) + ef(x-3h)$ Note the ordering of the coefficients. The coefficients are the same as for above:

$$a = -\frac{11}{12h^2}; b = \frac{-5}{3h^2}; c = \frac{1}{2h^2}; d = \frac{1}{3h^2}; e = -\frac{-1}{12h^2}$$

4. **Centered Three Point Stencil for First Derivative**

Approximating double derivative as $f'(x) = af(x-h) + bf(x) + cf(x+h)$, we wish to find the coefficients and find an order of approximation for the error.

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) + O(h^3)$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + O(h^3)$$

$$f'(x) = af(x-h) + bf(x) + cf(x+h)$$

$$\Rightarrow f'(x) \approx (a+b+c)f(x) + (-a+c)hf'(x) + \frac{a+c}{2}h^2f''(x)$$

$$a+b+c=0$$

$$-ah+ch=1$$

$$\frac{a}{2} + \frac{c}{2} = 0$$

$$\therefore a = -1/2h; b = 0; c = 1/2h$$

$$f'(x) = af(x-h) + bf(x) + cf(x+h)$$

$$= -1/2hf(x-h) + 1/2hf(x+h) + O(h^2)$$

5. **Centered Three Point Stencil for Second Derivative**

Following similar procedure as above with $f''(x) = af(x-h) + bf(x) + cf(x+h)$, we get

$$f''(x) = af(x-h) + bf(x) + cf(x+h)$$

$$= 1/hf(x-h) + -2/hf(x) + 1/hf(x+h) + O(h^2)$$

2.3 More on Error

We used Taylor's theorem to use find the error in our approximations to derivatives. However, this is only the theoretical error. Generally we get more accurate results as we decrease the step size 'h'. However, as h gets smaller, we have to account for numerical error and be wary of the phenomena of rounding error and catastrophic cancellation. We can derive a bound on the numerical error by doing a forward error analysis, i.e. by keeping track of rounding error in each arithmetic calculation and taking the worst case errors. The bound will be a function of ε_{mach}/h^n for some positive integer n. We can therefore find an optimal 'h' value

for which our calculation will be the most accurate for a calculation of a derivative. We just need to equate our numerical error with our theoretical error due to Taylor sum truncation, and solve for h .

We could derive an optimal h step size for the case of a centered three point stencil approximation for the second derivative. But, note that our project involves eigenvalue calculation and linear system solving for $1/h$ by $1/h$ size matrices (see below). To save computing time, we will not go beyond $h = O(10^{-3})$ for our step sizes, and therefore error in our calculation will be dominated by the theoretical error due to truncation of Taylor sum. The numerical error dominates only for very small h .

3 Solving the Schrodinger's Equation

3.1 Eigenvalue Problem

The wavefunctions, u , are eigenvectors of the Hamiltonian operator, and satisfy the Schrödinger Equation:

$$\hat{H} u = E u \quad (1)$$

where \hat{H} is the Hamiltonian Operator, and the eigenvalues E are the energies of a particle with wavefunction u .

In the one dimensional case, u is dependent only on the spatial coordinate x , and the one dimensional Hamiltonian is given by

$$\hat{H} = \left(\frac{-\hbar}{2m} \right) \frac{\partial^2}{\partial x^2} + V(x)$$

where \hbar is a constant, m is the mass of the particle, and $V(x)$ defines the potential energy function for the particle. Note that \hat{H} is dependent upon x via the $V(x)$ term. To give a dimensionless analysis, we define $\hbar = 2m = 1$, so that

$$\hat{H} = -\frac{\partial^2}{\partial x^2} + V(x) \quad (2)$$

Consider the points: $x_j \quad j = 0, 1, 2 \dots n$.

At each point x_j , equation (1) holds, so that

$$\hat{H}_j u(x_j) = E u(x_j)$$

where \hat{H}_j is the Hamiltonian operator evaluated at $V(x_j)$. Considering each of the points x_j gives the system of equations:

$$\begin{aligned} \hat{H}_0 u(x_0) &= E u(x_0) \\ \hat{H}_1 u(x_1) &= E u(x_1) \\ &\dots \\ \hat{H}_n u(x_n) &= E u(x_n) \end{aligned}$$

Using equation 2, this system can be written as:

$$\begin{aligned} -u''(x_0) + V(x_0)u(x_0) &= Eu(x_0) \\ -u''(x_1) + V(x_1)u(x_1) &= Eu(x_1) \\ &\dots \\ -u''(x_n) + V(x_n)u(x_n) &= Eu(x_n) \end{aligned} \quad (3)$$

We now have a system of n equations relating the wavefunction of a particle to its energy. Given a potential function $V(x)$, we wish to evaluate the wavefunction to high accuracy at each of the sample points $u(x_j)$. We will limit our analysis to potential functions that go to infinity at the boundaries a and b of our grid, and will assume that the points x_j are equally spaced.

Note that this system consists of n solutions to the second order ODE

$$-u''(x) + [V(x) - E]u(x) = 0$$

This will require two boundary conditions which will vary by the case being analyzed. We will limit our analysis the evaluation of $u(x_j)$ in the case of the infinite square well, and for scattering states within a finite interval.

We now choose to approximate $u''(x)$ using a using some finite difference scheme for $u''(x)$, such that:

$$u''(x_j) = \left(\frac{1}{h^2}\right) (Au(x_{j-n}) + Bu(x_{j-n+1}) + \dots + Cu(x_j) + \dots + Du(x_{j+n-1}) + Eu(x_{j+n}))$$

where the coefficients A, B, C, D, and E are constants (that may be equal to zero), h is the step size, and n is an integer.

For the second order, centered finite difference approximation:

$$u''(x_j) \approx \left(\frac{1}{h^2}\right) (u(x_{j-1}) - 2u(x_j) + u(x_{j+1}))$$

system (3) becomes

$$\begin{aligned} & -\left(\frac{1}{h^2}\right)u(x_{-1}) + \left(V(x_0) + \frac{2}{h^2}\right)u(x_0) - \left(\frac{1}{h^2}\right)u(x_1) = Eu(x_0) \\ & -\left(\frac{1}{h^2}\right)u(x_0) + \left(V(x_1) + \frac{2}{h^2}\right)u(x_1) - \left(\frac{1}{h^2}\right)u(x_2) = Eu(x_1) \\ & \dots \\ & -\left(\frac{1}{h^2}\right)u(x_{n-2}) + \left(V(x_{n-1}) + \frac{2}{h^2}\right)u(x_{n-1}) - \left(\frac{1}{h^2}\right)u(x_n) = Eu(x_{n-1}) \\ & -\left(\frac{1}{h^2}\right)u(x_{n-1}) + \left(V(x_n) + \frac{2}{h^2}\right)u(x_n) - \left(\frac{1}{h^2}\right)u(x_{n+1}) = Eu(x_n) \end{aligned}$$

Substituting the approximation for $u''(x)$ into system (3) gives a system of n linear combinations relating each $u(x_j)$ and nearby $u(x_k)$ to $Eu(x_j)$. We define the vector

$$\vec{u} = \begin{bmatrix} u(x_0) \\ u(x_1) \\ \dots \\ u(x_{n-1}) \\ u(x_n) \end{bmatrix}$$

By using an appropriate finite differencing scheme, and utilizing boundary conditions, the system of linear equations may be expressed only in terms of the sample points $u(x_j)$ $j = 0, 1, \dots, n$

(i.e. excluding $u(x_{-1}), u(x_{-2}), \dots$ and $u(x_{n+1}), u(x_{n+2}), \dots$).

In this case, system (3) may be written as

$$H\vec{u} = E\vec{u} \quad (4)$$

where H is a matrix containing the coefficients of each $u(x_j)$ in the system of linear equations. Here, E is an eigenvalue of H , and the energy of the eigenvector \vec{u} , which corresponds to the value of a wavefunction $u(x)$ taken at sample points x_j .

Note that E is simply a scalar, and equation (4) may be alternatively written as

$$H_E\vec{u} = 0 \quad (5)$$

where H_E is simply H with E subtracted from its diagonal entries.

Recall that equations (4) and (5) require defining the linear system only in terms of $u(x_j)$ where $j = 0, 1, \dots, n$. We now examine how this condition may be met for the infinite square well, and scattering states in a finite interval.

fivepointschrod1: This function solves for the H matrix for the centered five point stencil problem. The first and the last rows of the fivepointschrod1.m are adjusted to use forward and backward five point implementation respectively that are also described in Section 3.2.

schrodcenter: This function takes in coefficients for any centered difference stencil in an array along with an array containing coefficients for forward/backward stencil for the first and last rows of H. This allows the infinite square well problem to be solved using arbitrary order finite difference schemes.

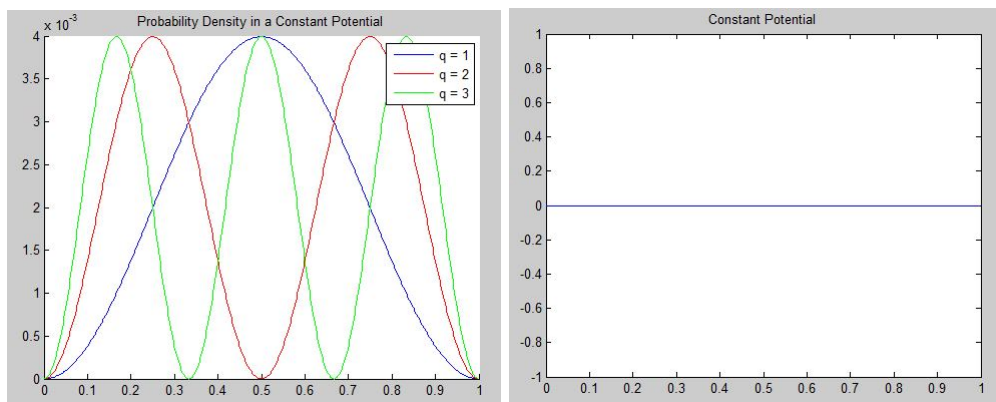
Potential Functions: We have several function handles for one dimensional potential functions. constpoten describes a constant potential of zero. barrier.m and describes a rectangular potential extending over [0.4,0.6], harmosc describes the linear harmonic oscillator, morse gives the morse potential, and varied gives a piece-wise potential of varying effects.

plotenergygraphs: We can find the eigenvalues and eigenvectors of H using the matlab command eig(H). This function then checks the calculated energies against the analytically calculated energy values. See below for explanation and plots.

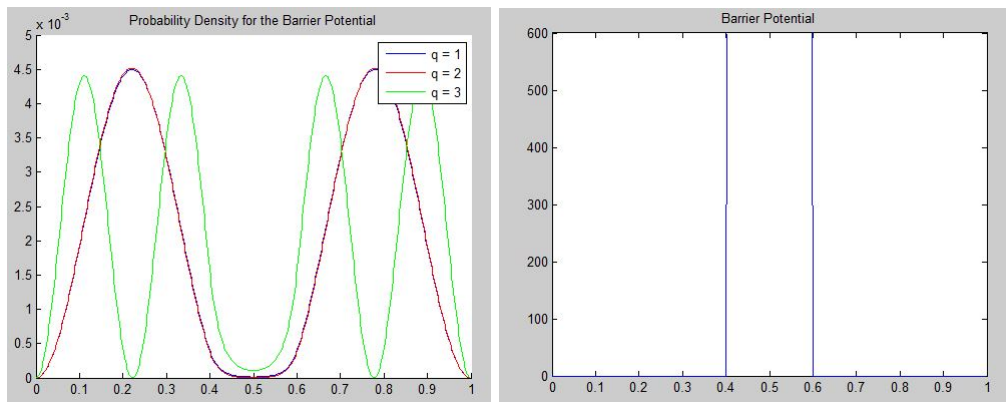
comparError: This compares convergence rates for both three point and five point stencils for a given quantum number q and a given range of values of grid points N.

4.2 Solution of $u(x_j)$ for Various Physically Relevant Potential Functions

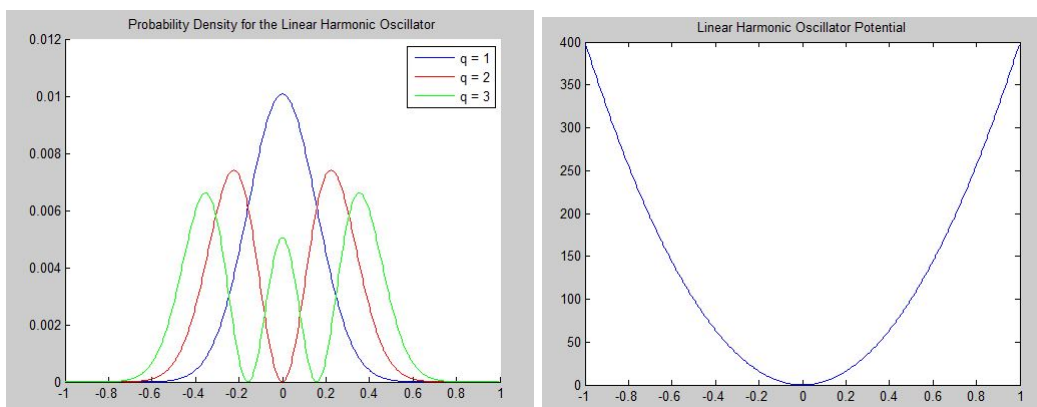
Constant Potential: The constant potential function is relevant because it allows for an analytic solution to the Schrödinger Equation . This will allow for error convergence analysis.



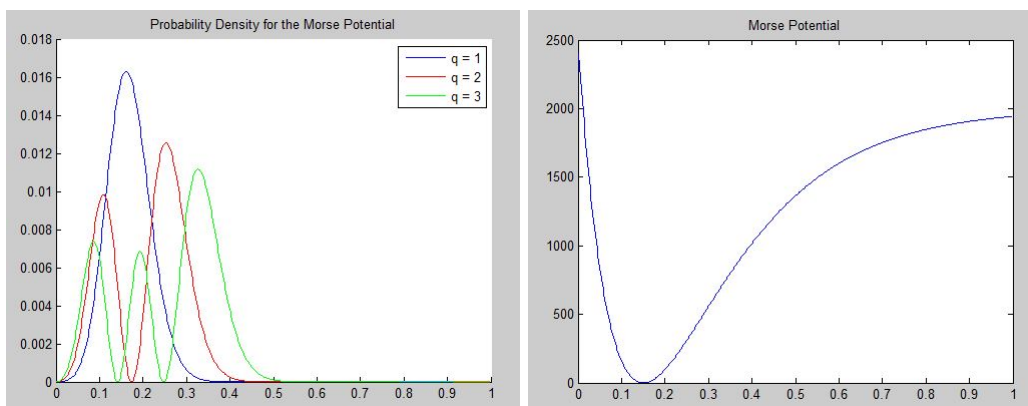
Rectangular Barrier: The rectangular barrier can be used to demonstrate the phenomenon of quantum tunneling.



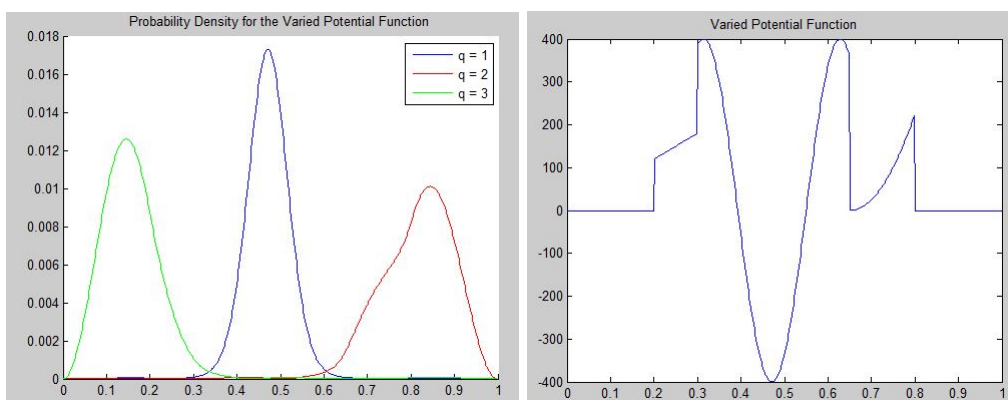
Linear Harmonic Oscillator: The Linear Harmonic Oscillator also gives an analytic solution to the Schrödinger Equation, and provides a good approximation for the vibrational states of a diatomic molecule.



Morse Potential: The Morse Potential provides a good approximation to the vibrational states of a molecule, and includes the anharmonicity inherent in chemical bonds.



Varied Potential: Because we can...



4.3 Error and Convergence

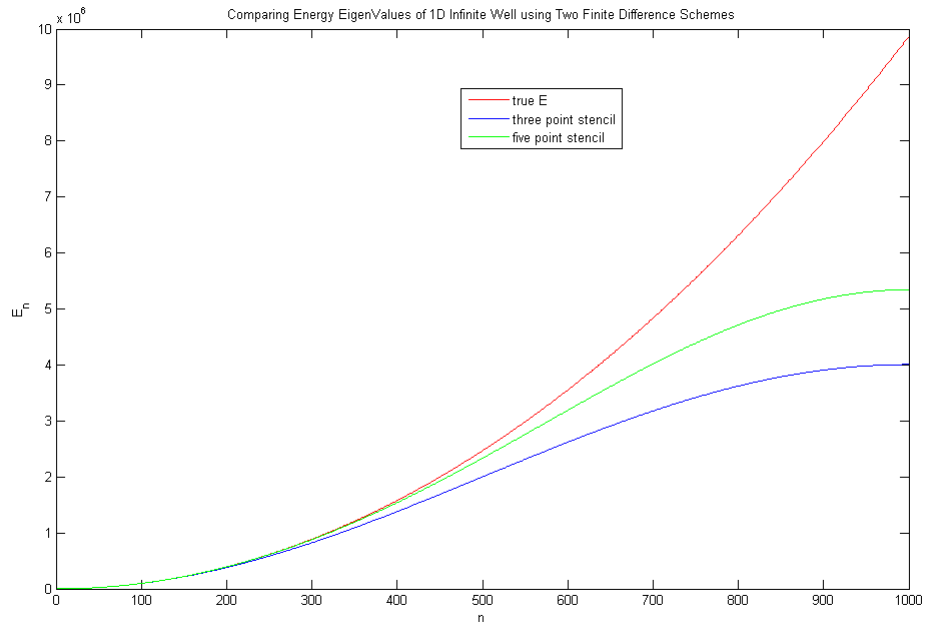


Figure 1: Comparing Energy Calculation for Different Stencils against True Answer.

Explanation:

Figure 1 shows the plot of energies using the three point stencil and the five point stencil. They are plotted against the true parabolic energy on the above figure; $N=1000$ grid points were used. Therefore energies were calculated for quantum numbers 1 to 1000. The finite difference schemes only follow the true parabolic shape for smaller quantum numbers. The five point stencil is good for a larger range of quantum numbers than the three point stencil.

This suggests that in using finite difference schemes to solve the schrodinger's equation, we are effectively solving for a slightly different problem than the actual quantum harmonic oscillator. This is consistent with the fact that we are truncating taylor sums to get approximations for the second derivative.

We can predict that the convergence of small quantum numbers for increasing no of grid points N will be close to our prediction of $O(N^{-2})$ for three point stencil and $O(N^{-4})$ for five point stencil. For larger quantum numbers, the convergence should be increasingly worse.

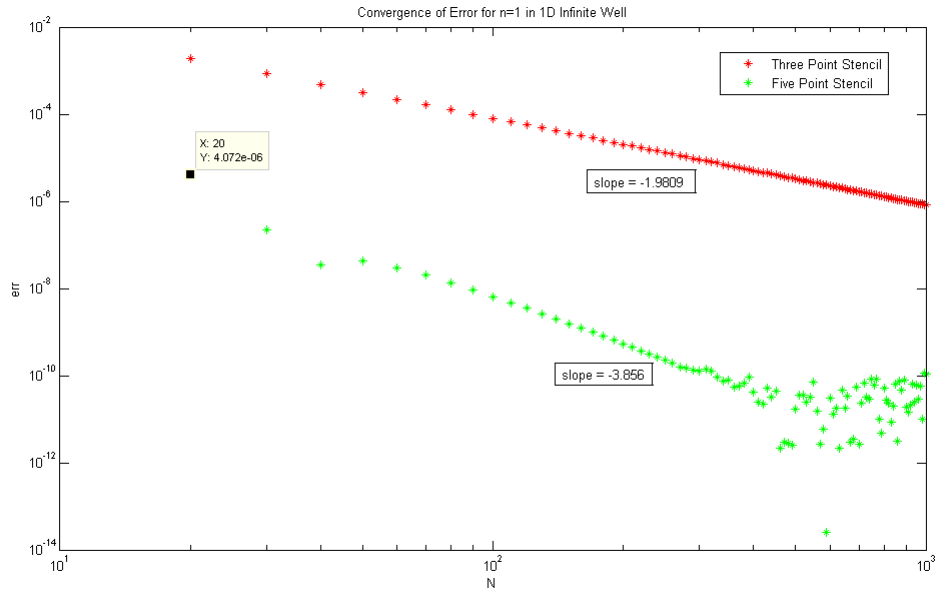


Figure 2: Convergence Rates for $q = 1$.

Explanation:

Figure 2 shows the convergence rates plot for $q=1$ quantum number. Using `comparError.m` code, we plot the error in the calculated energies for each stencil against decreasing step size (h as described in Section 3.2), i.e. against increasing grid points N . First we notice that the five point stencil is significantly better than the three point stencil. We get 4 digits of accuracy using a little more than $N=50$ grid points for the five stencil case while the three stencil case requires more than $N=500$ grid points for 4 digits of accuracy. We also notice that the slope of the plots (which are both loglog plots implemented in matlab) show that the convergence matches our prediction for both 3 point and 5 point stencils. However, we see that five point stencil is a little less than order 4 algebraic convergence. This can be attributed to the constant in the error term as described in Section 2.3 during the derivation of the error for the centered five point stencil. Also for $N > 450$ grid points, we reach more than 10 digits of accuracy and the numerical calculation does not seem to give more digits of accuracy since the section of the plot for $N > 450$ in the case of 5 point stencil is not following a trend. We attribute this to the fact that there are numerical errors along with the truncation theoretical error in approximating second derivatives, as described in Section 2.3.

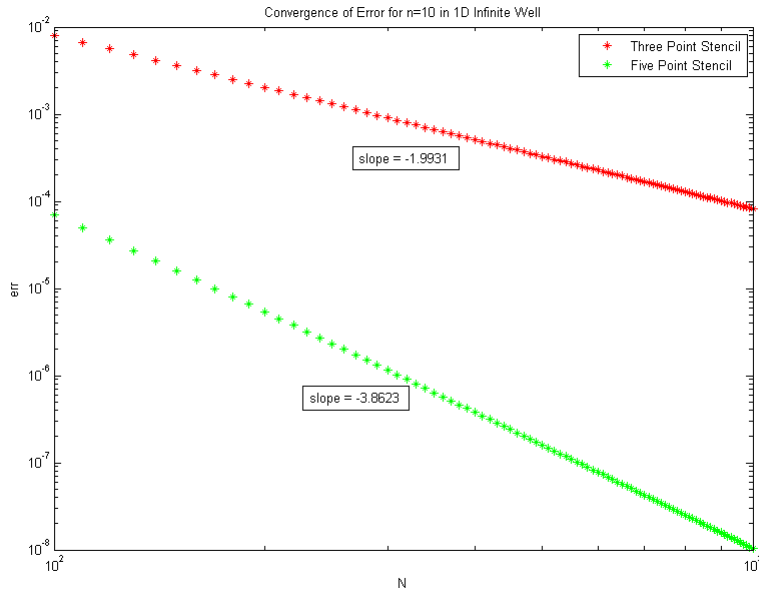


Figure 3: Convergence Rates for n=10 Quantum Number.

Explanation:

Figure 3 shows the convergence rates for the $q=10$ quantum number. Again, the five point stencil is much better giving us 8 digits for $N=1000$ grid points. The three point stencil gives us about 4 digits in as many grid points. This agrees with the analytically predicted convergence rates for these two stencils.

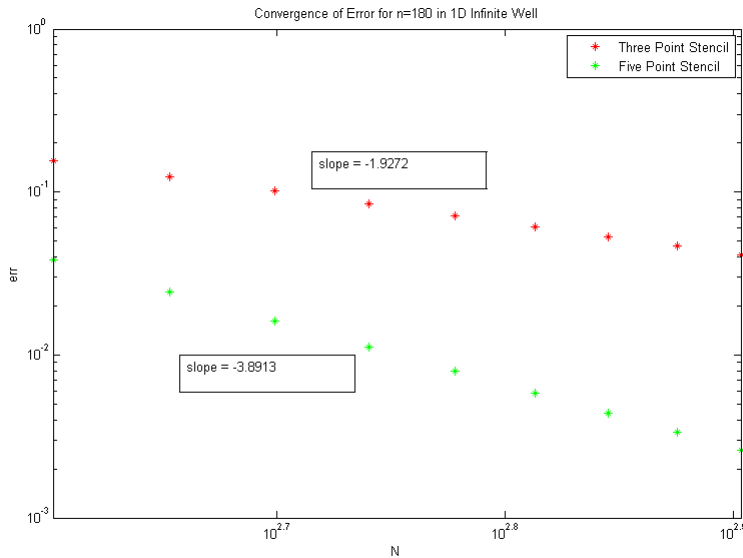


Figure 4: Convergence Rates for n=180 Quantum Number.

Explanation:

As we move on to higher quantum numbers, i.e. as we approach the region where the computed energies diverge from the analytic energies, we see there are fewer digits of accuracy. In the above plot, we have 2 to 3 digits of accuracy for the five point stencil while only 1 digit of accuracy for the three point stencil. Note that the convergence rates are decreasing as quantum number increases.

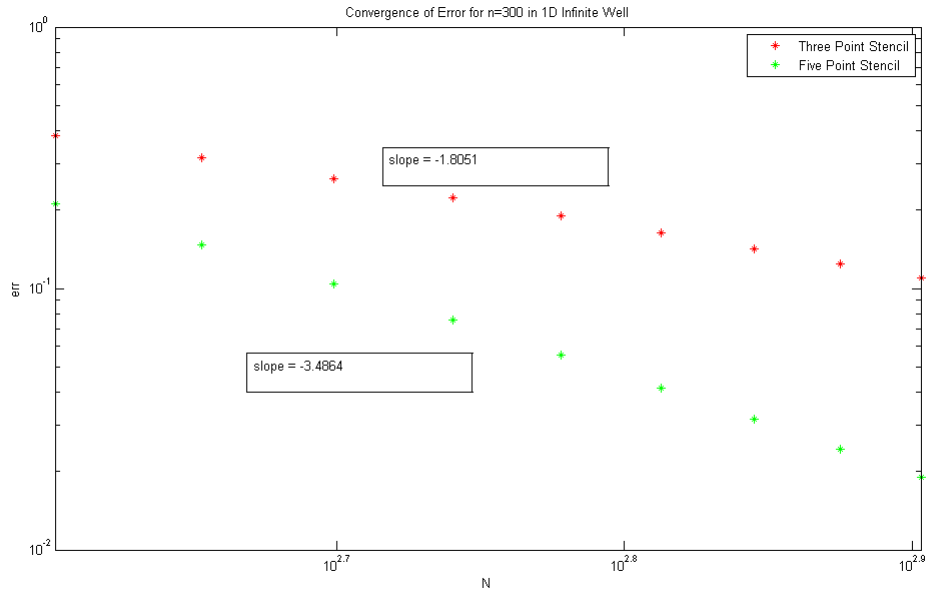


Figure 5: Convergence Rates for n=300 Quantum Number.

Explanation:

At this point, when we are barely getting 1 digit of accuracy, and convergence is very poor. Neither the three point stencil nor the five point stencil give accurate results, and either more sample points, or a different method, are required.

4.4 Runtime

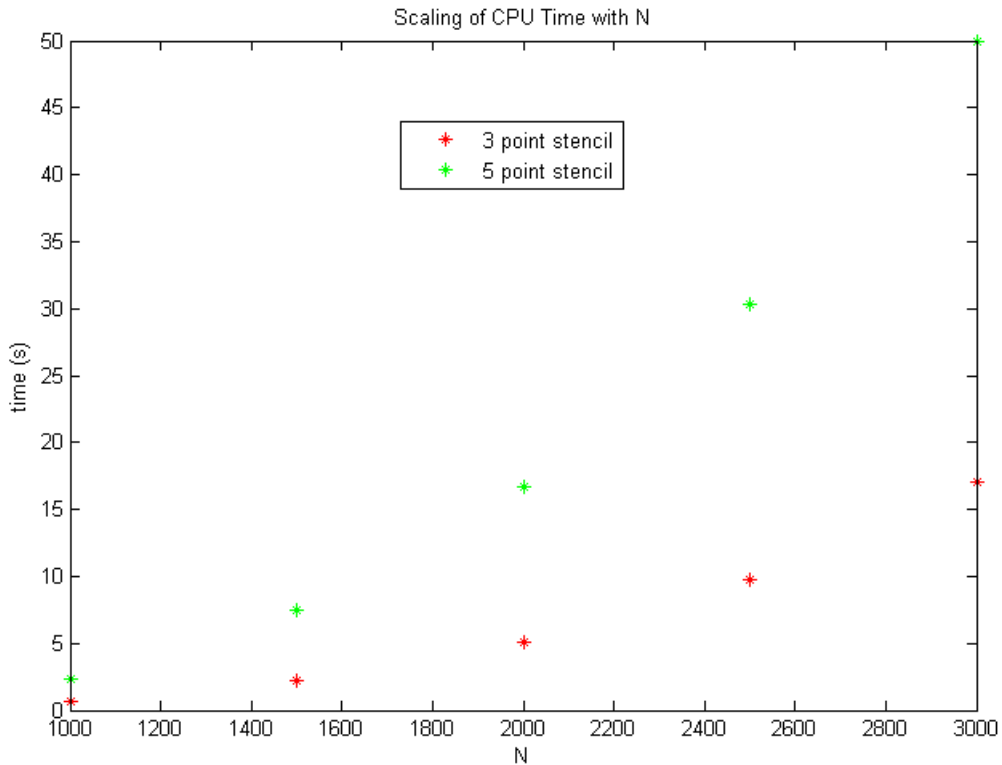


Figure 6: CPU Time Scale with Increasing N.

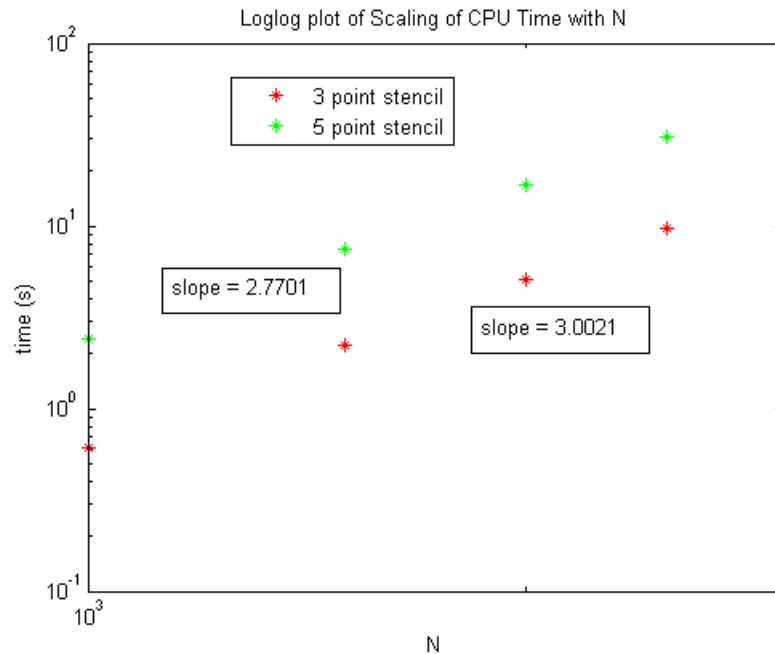


Figure 7: Log Log plot of CPU Time Scale with Increasing N.

Explanation:

In general, it takes longer to calculate eigenvalues of H using the 5 point stencil than using the 3 point stencil. However, as the loglog plot of cpu time vs N, the number of sample points, demonstrates (Figure 7), the scaling with N for the 5 point stencil is slightly smaller than for the 3 point stencil.

4.5 Eig vs Eigs

Since H is a sparse matrix, we can use the MATLAB function eigs to find the eigenvalues much faster for larger numbers of grid points N. While eig can only find eigenvalues quickly for smaller N values, eigs can find a few eigenvalues of a sparse matrix for very large N values very quickly. We use this section to find convergence plots for small quantum numbers using eigs. To do so, we first have to make MATLAB recognize that H is a sparse matrix by using the function sparse(H). Then we can use the command eigs(sparse(H),5,0) to find the first five eigenvalues for the infinite square well. The MATLAB code where eigs is implemented is called eigsconvergence.m. Here, we demonstrate the usefulness of eigs by getting more digits of accuracy for q=1 quantum number using the three point stencil.

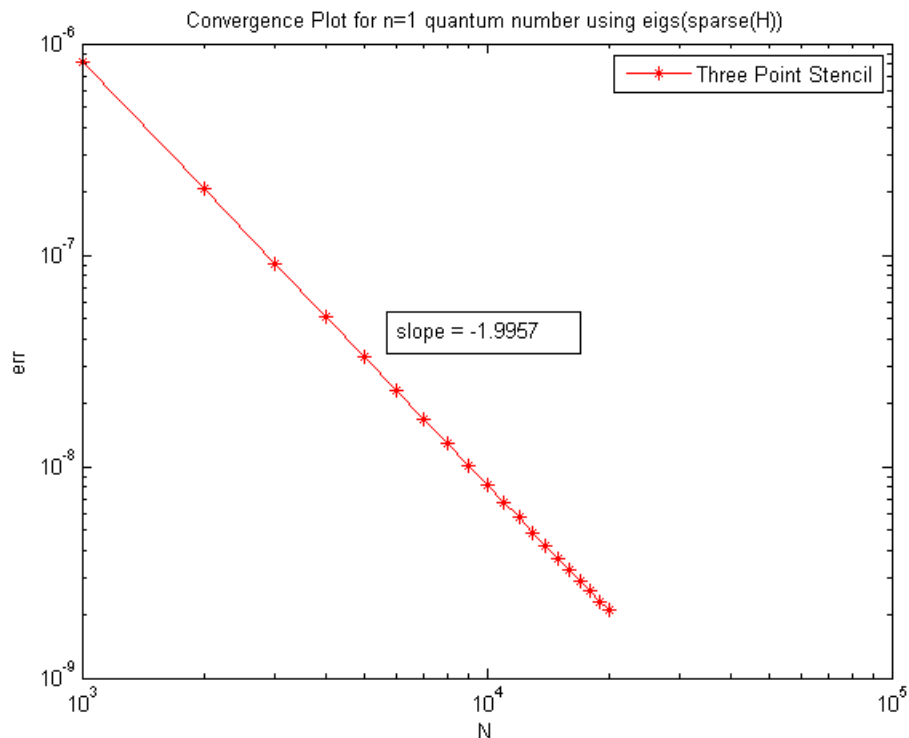


Figure 8: Convergence check using N from 1,000 - 20,000 grid points.

Explanation:

While we were able to get accuracy to only about 5 digits by using N upto 1000 grid points before (see Figure 2) using eig, we see in Figure 8 above that we are able to get up to about 9 digits using N upto 20,000 grid points. Since eigs only calculates a few eigenvalues from the sparse matrix H, the computing time is not that long either. This suggests that we can use eigs if we are concerned with only a few smallest or largest eigenvalues to get more accurate results.

5 Scattering Problem

For the scattering problem, the particle may be found in an infinitely large interval. The wavefunction of a completely free particle may be expressed as a traveling wave, or

$$u(x) = Ae^{-ikx}$$

When a potential barrier is present, a traveling wave incoming from the left of the potential will partially pass over the barrier and continue as an altered traveling wave, and part of the wave will be reflected off the barrier and travel in the direction of its origin. Thus, a compact support potential function over the interval [a,b] will divide the dimension into three regions: Region 1 is $(-\infty, a)$, Region 2 [a,b], and Region 3 (b, ∞) .

In Region 1, the wavefunction may be expressed as

$$u_1(x) = Ae^{-ikx} + Be^{ikx}$$

where $k = \sqrt{E - V_0}$. This may be considered as an incoming traveling wave Ae^{-ikx} from the left, and a reflected wave Be^{ikx} traveling to the right.

In Region 2, $u_2(x)$ is unknown and depends upon the nature of the potential barrier.

In Region 3, the particle is quasi-free, and $u_3(x)$ consists only of an outgoing traveling wave

$$u_3(x) = Ce^{-ikx}$$

moving to the left.

Since the wavefunction is unknown in the Region 2, we wish to solve for the values of $u_2(x_j)$ using approximate, finite-difference methods. Given the approximate solution, the equations of $u_1(x)$ and $u_3(x)$ may be determined, and $u_{1,3}(x_k)$ computed directly.

For this case, we define the x_j so that $x_0 = a$ and $x_n = b$. As shown in section 3.1, solving for the values of $u_2(x_j)$, which hereafter will be referred to as $u(x_j)$, requires system (3) to consist of linear combinations of only the sample points $u(x_j)$ for $j = 0, 1, \dots, n$. Thus, for Region 2, the system must consist of linear combinations of $u(x_j)$ where $x_j \in [a, b]$.

For the second order, centered approximation of $u''(x)$, each linear combination may be expressed as:

$$-\left(\frac{1}{h^2}\right)u(x_{j-1}) + \left[V(x_j) - E + \frac{2}{h^2}\right]u(x_j) - \left(\frac{1}{h^2}\right)u(x_{j+1}) = 0$$

These combinations satisfy the condition of equation (5), except for x_0 , and x_n , which depend on the terms $x_{-1} \notin [a, b]$ and $x_{n+1} \notin [a, b]$.

For x_0 and x_n we require the boundary conditions for the system.

Noting that $u_1(a) = u_2(a)$ and $u_3(b) = u_2(b)$, the boundary conditions are

$$\begin{aligned} u_2(x_0) &= Ae^{-ikx} + Be^{ikx} \\ u_2(x_n) &= Ce^{-ikx} \end{aligned}$$

For a dimensionless approximation, set $A = 1$ so that the boundary conditions are:

$$\begin{aligned} u_2(x_0) &= e^{-ikx_0} + Be^{ikx_0} \\ u_2(x_n) &= Ce^{-ikx_n} \end{aligned}$$

Since B and C are unknown, this form of the boundary conditions is not particularly useful. However, taking the derivative of each condition gives:

$$\begin{aligned} u'_2(x_0) &= -ike^{-ikx_0} + ikBe^{ikx_0} = -ike^{-ikx_0} + iku_2(x_0) - ik e^{-ikx_0} = -2ike^{-ikx_0} + iku_2(x_0) \\ u'_2(x_0) - iku_2(x_0) &= -2ike^{-ikx_0} \\ \text{and} \\ u'_2(x_n) &= -ikCe^{-ikx_n} = -iku_2(x_n) \\ u'_2(x_n) + iku_2(x_n) &= 0 \end{aligned}$$

A finite difference scheme may be used to represent $u'_2(x_0)$ and $u'_2(x_n)$ in terms of $u_2(x_j)$.

Since, x_0 is the rightmost point in $[a, b]$, the scheme for $u'_2(x_0)$ can only involve $u_2(x_j)$ to the right of $u'_2(x_0)$, so a forwards scheme must be used.

Similarly, x_n is the leftmost point in $[a, b]$, so a backwards finite difference scheme must be used for $u'_2(x_n)$. For the first order, forward and backwards schemes for $u'(x)$:

$$\begin{aligned} u'(x_0) &\approx \left(\frac{1}{h}\right)(-u(x_0) + u(x_1)) \\ u'(x_n) &\approx \left(\frac{1}{h}\right)(-u(x_n) + u(x_{n-1})) \end{aligned}$$

The two linear equations are:

$$\begin{aligned} \left(-\frac{1}{h} - ik\right)u(x_0) + \frac{1}{h}u(x_1) &= -2ike^{-ikx_0} \\ \left(\frac{1}{h}\right)u(x_{n-1}) + \left(ik - \frac{1}{h}\right)u(x_n) &= 0 \end{aligned}$$

This gives a system of equation consisting of linear combinations of only $u(x_j)$, such that:

$$H_S \vec{u} = \vec{b}$$

where the matrix H_S is given by:

$$\begin{pmatrix} -\frac{1}{h} - ik & \frac{1}{h} & 0 & \dots & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V(x_1) - E & -\frac{1}{h^2} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V(x_{n-1}) - E & -\frac{1}{h^2} \\ 0 & \dots & 0 & \frac{1}{h} & ik - \frac{1}{h} \end{pmatrix}$$

and \vec{b} is a vector of all zeros, except for the first entry, which is $-2ike^{-ikx_0}$. Obtaining \vec{u} is a simple matter of computing $H_S^{-1}\vec{b}$.

5.1 MATLAB Implementation

The functions `scatter` and `scatter2` solve the scattering problem for \vec{u} given a potential function, number of sample nodes, and desired interval. Additionally, since the values of E are no longer quantized, an input value for E is required.

scatter: The function `scatter` uses a first order forward/backward finite difference scheme of $u'(x)$ to represent the boundary cases of the matrix H_S and a second order centered scheme for $u''(x)$ for all other cases.

scatter2: The function `scatter2` uses a second order forward/backwards finite difference scheme for $u'(x)$ to represent the boundary cases of the matrix H_S , and a second order centered scheme for $u''(x)$ for all other cases.

converge: The function `converge` was used for the convergence analysis of `scatter`.

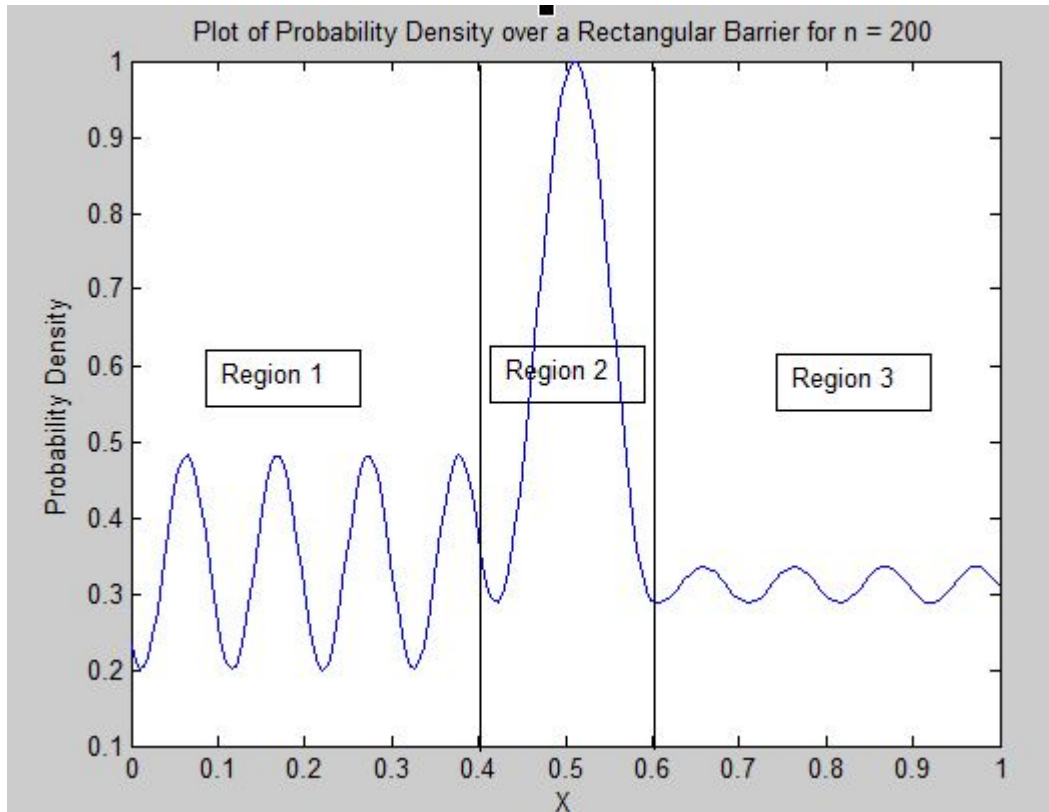
converge2: The function `converge2` was used for the convergence analysis of `scatter2`.

5.2 Convergence of scatter

Since the wavefunction in region 3 is described by a single traveling wave:

$$u_3 \cdot u_3^* = C$$

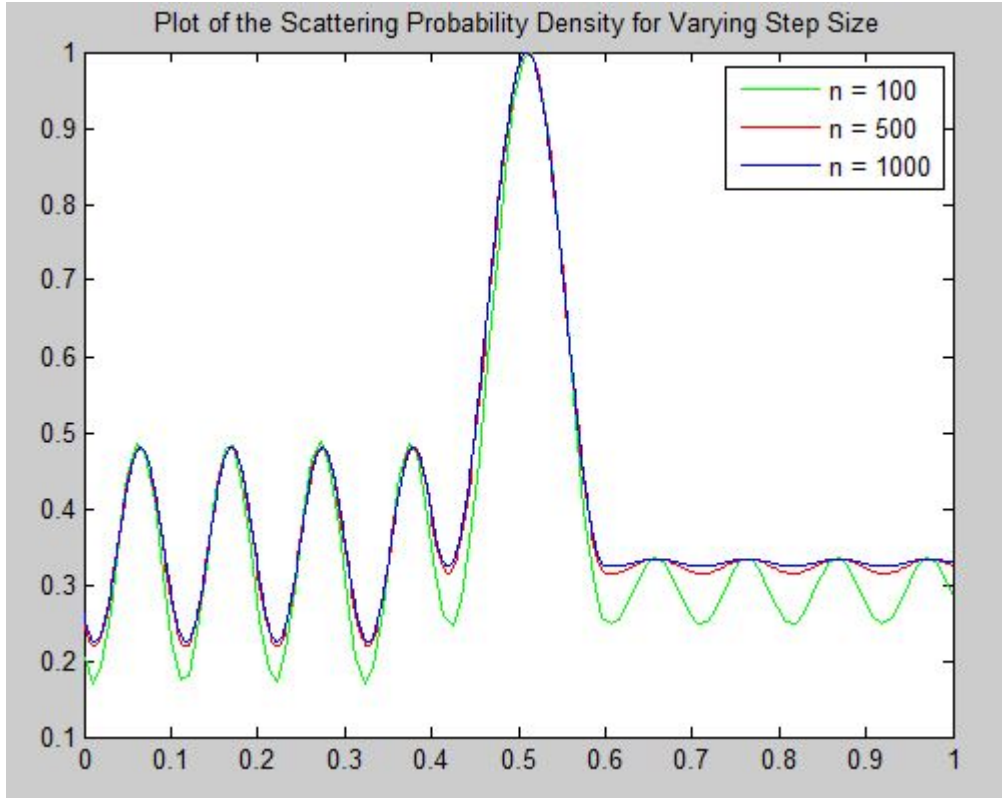
where C is some constant. This is true for any compact support potential function. However, plotting $u_3 \cdot u_3^* = P$ for any potential gives a plot similar to the following:



In this case, a square barrier extends over $[0.4, 0.6]$, the energy of the particle was 900. Let this be known as test case 1. This plot is for 200 sample points.

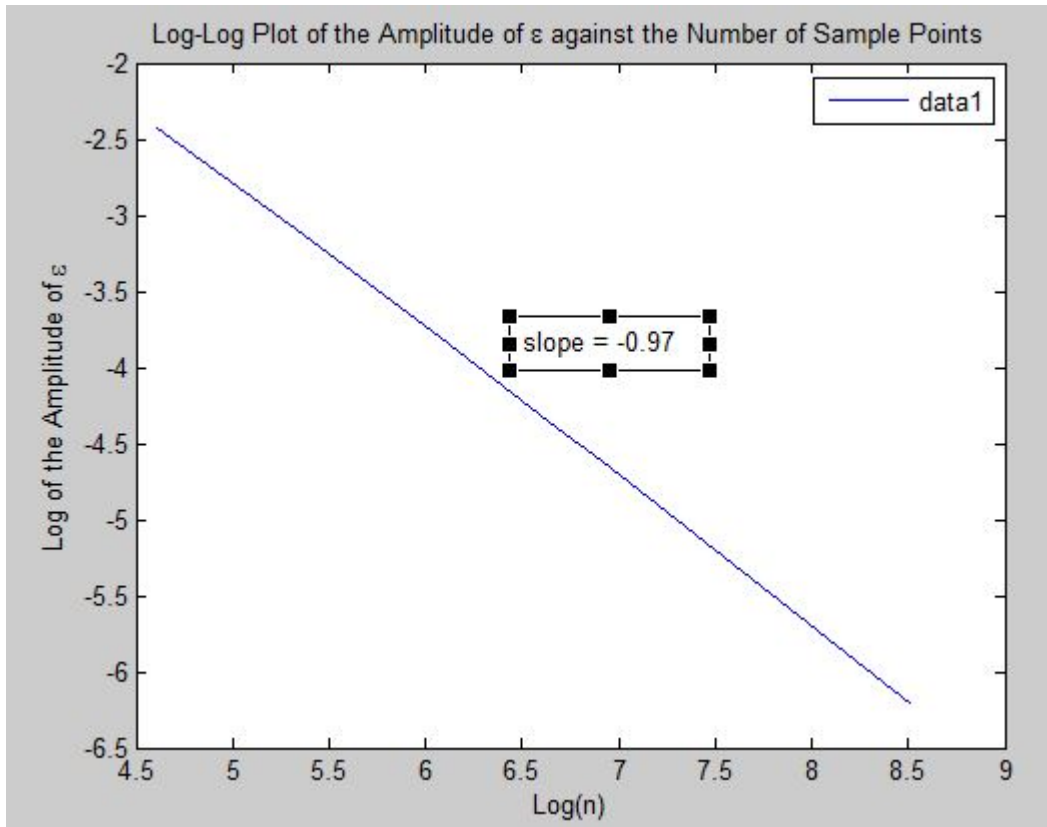
We observe experimentally that P is periodic in region 3, rather than a constant. This effect holds for any choice of potential function, energy, step size, and interval. We hypothesize that the amplitude of the error wave in region 3 could serve as a test of the convergence of the function.

For finite-difference schemes, convergence of various orders depends on the spacing between the sample points. If the amplitude of P is to conform with this behavior, then the amplitude of the error wave ϵ should decrease as step size decreases. For test case 1 with increasing number of sample points (and therefore decreasing step size), we obtain the following result:



As step size decreases, the amplitude of the error wave decreases, and the wavefunction in region 3 approaches a constant value. Thus, ϵ satisfies our requirements for a test of convergence.

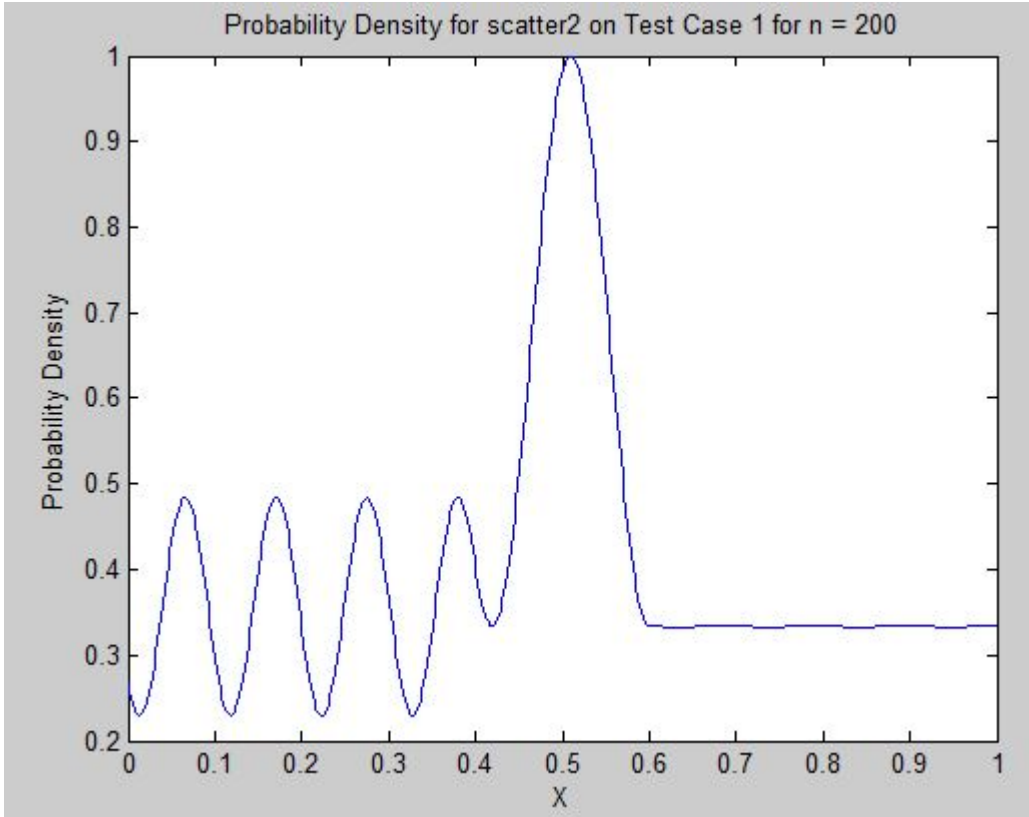
Using the MATLAB code converge, the amplitude of ϵ was extracted for test case 1 for varying n, and the following log-log plot obtained:



Note that this is algebraic convergence of order 1, the analytically expected convergence for a first order finite difference scheme.

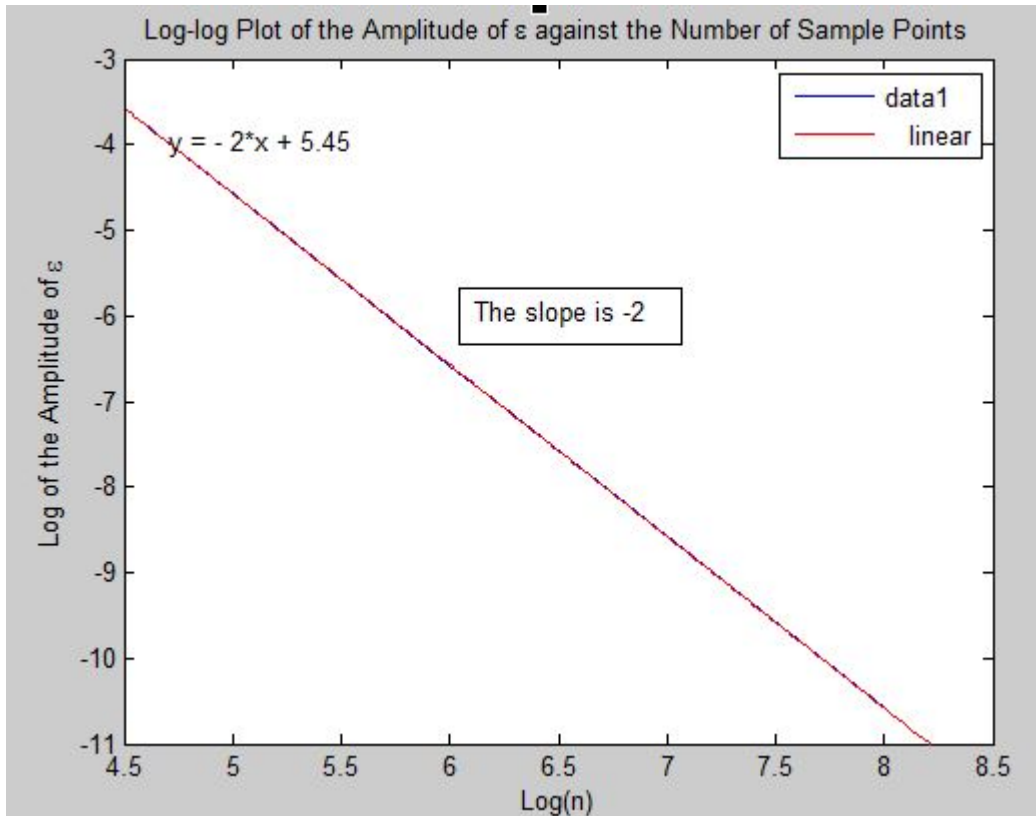
5.3 Convergence of scatter2

The function scatter2 uses a second order finite difference scheme for the boundary conditions of H_S , so algebraic convergence of order 2 is expected. For test case 1 and $n = 200$, we obtain the following plot of probability density:



Note that the magnitude of the error wave ϵ is noticeably smaller than for the analogous plot for scatter. Since scatter2 uses a second order finite difference scheme, as opposed to the first order finite difference scheme of scatter, this is an expected result.

Using converge2, the amplitude of ϵ for scatter2 and test case 1 for varying step sizes was extracted, and the following log-log plot obtained.



Note that this is algebraic convergence of order 2, the analytically expected result for a second order finite difference scheme.

6 Conclusion

We conclude that finite difference schemes offer an easy way to solve the 1D Schrodinger's Equation numerically. Generally, the results are more accurate for higher order finite difference methods. For example, for the infinite well problem, we obtained more accurate results with a small number of grid points by using the five point stencil than by using the three point stencil. However, we also observed that the computed energies diverged from the analytic solution for high quantum numbers.

We confirmed the predicted effects of using smaller step sizes in approximating derivatives numerically in Section 4.3. We saw that the convergence of error using the three point stencil was $O(N^{-2})$ and the convergence of the five point stencil was $O(N^{-4})$.

We also found that we can utilize MATLAB's eigs command to solve the eigenvalues of the sparse matrix H. In the scattering problem, we found that the error in the probability density takes the form of a wave, and the amplitude of this wave corresponded to the convergence of the numerical solution. Decreasing the step size decreased the amplitude of this wave, with a first order finite difference scheme converging $O(N^{-1})$ and a second order scheme converging by $O(N^{-2})$ as analytically predicted.

7 Sources

1. McQuarrie, Donald A. Quantum Chemistry. Mill Valley, CA: U Science, 1983. Print.
2. Griffiths, David J. Introduction to Quantum Mechanics. Upper Saddle River, NJ: Pearson Prentice Hall, 2005. Print.
3. <http://faculty.washington.edu/rjl/fdmbook/>

4. <http://www-personal.umich.edu/~lunmeih/math471fall12/project1.pdf>
5. <http://www.itp.phys.ethz.ch/education/fs08/cqp/cqp.pdf>